Acceptance of Inputs from Various Interfaces to a Telephony System

Inventor:
Michael S.W. Tovino

Technical Field

[0001] This invention relates generally to distributed telephony, and more particularly, to servers that accept client inputs from various interfaces and devices.

Background

[0002] Enterprises often have several offices or call centers that are located in a plurality of locations. To interconnect all of these sites, enterprise telephony systems have been developed. Enterprise telephony systems, which consist of a distributed set of voice switches and servers, offer enterprise applications enabled by the integration of computer systems with telephony services. The software that supports the computer-integrated functionality is generally implemented as a client-server environment in which the participants or clients (distributed telephony users) communicate directly with the server. Computer-integrated features rely not only on a server's application platform but also on the availability of the network that connects the switches, server, and application services.

[0003] Account codes are one example of an enterprise application. Account codes enable an enterprise to restrict phone calls made by employees and to associate phone calls with various entities, such as customers. This is accomplished by requiring a

telephony user to enter an account code before his call can be completed. Once a phone call has been associated with an entity, the enterprise can use this information, for example, to track time spent communicating with customers or to bill customers for phone calls. Depending on the situation, a phone call may require an account code or an account code may be optional.

[0004] In general, account codes are implemented as follows: A user initiates a phone call, thereby generating a phone call request. A server receives the request and determines whether an account code is necessary or optional for the phone call. If an account code is necessary or optional, the server prompts the user for an account code. In addition, if an account code is necessary, the server flags the phone call accordingly. The phone call is completed once the user has entered a valid account code. The phone call may also be completed if an account code is optional and the user has declined to enter an account code.

[0005] A user may initiate a phone call using any of a variety of devices. Each of these devices is different and has its own advantages and disadvantages for the user to initiate a conversation. These devices include both endpoints, such as analog phones, Internet-Protocol (IP) based phones, and software phones (softphones), and software applications that control these endpoints. These software applications, sometimes known as call managers, run on computers and offer users an interface through which the users may perform call-related functions such as transferring calls, placing calls on hold, and obtaining caller ID information, in addition to initiating calls.

[0006] Assume that a user initiates a phone call using a first device, such as an endpoint that will be used during the phone call. Traditionally, if the user wants to enter an account code, the account code must be entered using the first device. However, the user may prefer to use a second device to enter the account code, such as a call manager that has a user interface that enables the user to enter a name associated with an account code (e.g., a customer name) rather than the account code itself. Similarly, a user who has initiated a call using a call manager may wish to enter an account code using an endpoint. Traditional enterprise telephony systems require a user to enter an account code using the same device that was used to initiate the phone call.

[0007] What is needed is a system and method that enables a user to enter an account code using a device other than the device that was used to initiate the phone call.

Summary of the Invention

[0008] Computer-integrated functionality is implemented using a server in a distributed telephony environment. The server includes a telephony management software (TMS) unit, a telephony application programming interface (TAPI) unit, and a computer-integrated functionality unit. The server is coupled to one or more endpoints, such as analog phones, IP-based phones, and software phones. The server is also coupled to one or more software applications that control these endpoints. These endpoints and software applications may be used to input information into the telephony system that will then be associated with a call. Examples of such information are account codes and authorization codes. The server is also coupled to a storage device that stores such information.

[0009] When a server receives a request to make a call, it determines whether information may be associated with the call. If information may be associated with the call, the server sends a signal to the device that initiated the call and a signal to the endpoint or software application that is associated with this device. The signals cause the device and the associated endpoint or software application to prompt a user for input. The user may then input the information using either the device or the associated endpoint, regardless of which was used to initiate the call. The server then receives the information and determines whether it is valid. If it is, the server attaches the information to the call and completes the call.

[0010] Further features of the invention, its nature, and various advantages will be more apparent from the accompanying drawings and the following detailed description.

Brief Description of the Drawings

[0011] The accompanying drawings illustrate several embodiments of the invention and, together with the description, serve to explain the principles of the invention.

[0012] FIG. 1 is an illustration of an exemplary distributed telephony system architecture according to one embodiment of the present invention having two sites.

[0013] FIG. 2 is a diagram illustrating an exemplary server architecture according to one embodiment of the present invention.

[0014] FIG. 3 is an illustration of one embodiment of the account code unit.

[0015] FIG. 4 illustrates a method performed by a server for handling a request to make a call, prompting for an account code, and completing the call according to one embodiment of the present invention.

[0016] FIGS. 5A, 5B, and 5C are graphical user interfaces of a call manager, according to one embodiment of the present invention.

Detailed Description of the Embodiments

[0017] The present invention is now described more fully with reference to the accompanying figures, in which several embodiments of the invention are shown. The present invention may be embodied in many different forms and should not be construed as limited to the embodiments set forth herein. Rather, these embodiments are provided so that this disclosure will be thorough and complete and will fully convey the invention to those skilled in the art.

[0018] One skilled in the art will recognize that methods, apparatus, systems, data structures, and computer readable media implement the features, functionalities, or modes of usage described herein. For instance, an apparatus embodiment can perform the corresponding steps or acts of a method embodiment.

[0019] For simplicity purposes, the invention is described in the context of inputting account codes into a telephony system. However, the invention may be used to input any type of information into a telephony system, such as authorization codes and numbers for navigating automated message systems.

A. System Architecture

[0020] FIG. 1 is an illustration of an exemplary distributed telephony system architecture according to one embodiment of the present invention having two sites. The illustrated embodiment includes a first site 100A and a second site 100B. As used herein, a site represents a grouping of resources. In the illustrated embodiment, each of the two sites 100A, 100B is communicatively coupled via a network 190. One skilled in the art will note that sites 100A, 100B can be physically distinct from each other or merely topology-related groupings that are not in physically distinct locations. The system architecture in FIG. 1 is used only by way of example. While FIG. 1 illustrates two sites, the present invention applies to any system architecture containing one or more sites.

[0021] The first site 100A includes a server 110, a switch 130A, two endpoints (analog phone 121A and IP phone 122), a device running a call manager software application 150, and a storage device 140. The switch 130A represents a Voice over Internet Protocol (VoIP) device to which a number of endpoints (e.g., telephone devices) can be coupled, such as analog phones, IP phones, and software phones. In the illustrated embodiment, the switch 130A is coupled to the network 190. The switch 130A is also coupled to the public switched telephone network (PSTN) 180 via an analog or digital trunk line (e.g., a T1 or E1 interface). In the illustrated configuration, the switch 130A provides an interface for calls originating from or terminating on the PSTN 180.

[0022] An endpoint enables a user to carry on a phone call. Although in the illustrated embodiment the first site 100A has two endpoints (one analog phone 121A and one IP phone 122), in other embodiments the first site 100A has different numbers of

endpoints and different types of endpoints, such as software phones (softphones). An endpoint is coupled to a switch 130, a server 110, or both. An endpoint has a user interface to send data to and receive data from a user. In one embodiment, this interface enables a user to specify an account code based on an account name and also enables a user to search for an account code or account name. A set of account codes and names to search may be accessed through the server 110 or it may be stored on the endpoint itself.

[0023] The analog phone 121A has, for example, a Telephone User Interface (TUI) that sends data through a speaker and receives data through a microphone and a keypad. The IP phone 122 has, for example, both a TUI and a graphical user interface that sends data through a display device associated with the IP phone 122. In one embodiment, the IP phone's graphical user interface also receives data from a touchscreen display device associated with the IP phone 122. A softphone (not shown) has, for example, a software application that runs on a computer and sends data through a display device and a speaker and receives data through a microphone, a keyboard, and a pointing device.

[0024] A device running a call manager software application 150, such as a computer, controls one or more endpoints with which it is associated. The call manager 150 offers a user an interface through which the user may perform call-related functions such as initiating calls, transferring calls, placing calls on hold, and obtaining caller ID information.

[0025] In one embodiment, the call manager 150 enables a user to enter an account code that will be associated with a phone call. FIGS. 5A, 5B, and 5C are graphical user interfaces of a call manager, according to one embodiment of the present invention. In

one embodiment, such as the embodiment shown in FIG. 5A, a user may enter an account code. In another embodiment, such as the embodiment shown in FIG. 5B, a user may choose an account code from several displayed account codes, account names, or pairs of account code and account names. If the list of possible account codes or account names is large, the user may filter the list by typing in one or more characters of the desired account code or account name, as illustrated in FIG. 5C. The device running the call manager 150 may access the set of account codes and names through the server 110 or it may store this information itself.

[0026] Although in the illustrated embodiment the first site 100A has only one call manager 150, in other embodiments the first site 100A has a different number of call managers 150. Also, more than one call manager 150 may control the same endpoint. The association between a call manager 150 and an endpoint that it controls is accessed through the server 110.

[0027] The server 110 is configured to implement features or functions of the present invention. The server 110 is coupled to the network 190 and may also be coupled to one or more endpoints, such as IP phone 122. The server 110 will be further discussed below with reference to FIGS. 2-4.

[0028] The storage device 140 contains account code information, including account codes and the names of customers associated with the account codes. In the illustrated embodiment, the storage device 140 is coupled to the server 110. In an alternate embodiment, the storage device 140 is coupled to the network 190.

[0029] One skilled in the art will appreciate that additional networking devices can be added to the first site 100A, for example, if needed to support additional endpoints, servers, or other systems. For example, the first site 100A may include a second switch and an edge router to couple the first site 100A to the network 190 and to provide local area connectivity for the first and second switches. One skilled in the art will also recognize that numerous configurations of switches and communications links are contemplated. For example, PSTN links can be coupled to multiple switches at several points within the topology and softswitches can also be used.

[0030] The second site 100B similarly includes an endpoint (analog phone 121B) and a switch 130B. The configuration of the second site 100B demonstrates that a server is not required for each site. Switch 130B of the second site 100B can be managed by server 110 that is illustrated in the first site 100A. A call can involve more than one switch. For example, a call that originates from the PSTN 180 and terminates on an endpoint that is communicatively coupled to switch 130B of the second site 100B involves two switches: switch 130A of the first site 100A and switch 130B of the second site 100B. In addition, each switch 130 may be managed by a different server 110.

[0031] In one embodiment of the present invention, the network 190 is a partially public or a wholly public network such as the Internet. The network 190 can also be a private network or include one or more distinct or logical private networks (e.g., virtual private networks or wide area networks). Additionally, the communication links to and from the network 190 can be wireline or wireless (i.e., terrestrial- or satellite-based transceivers). In one embodiment of the present invention, the network 190 is an IP-based wide or metropolitan area network.

B. Server Architecture

[0032] FIG. 2 is a diagram illustrating an exemplary server architecture according to one embodiment of the present invention. In this embodiment, server 110 is configured to implement features or functions of the present invention. Server 110 includes a processor 210. The processor 210 can be a conventional processing device, such as a general-purpose microprocessor.

[0033] Server 110 also includes a memory 220. The memory 220 includes program instructions or functional units that implement features of the present invention. Specifically, the memory 220 includes a telephony management software (TMS) unit 230 and a telephony application programming interface (TAPI) unit 240.

[0034] In one embodiment, the memory 220 also includes one or more application units that interact with the TMS unit 230 and the TAPI unit 240 to enable a specific computer-integrated function. An application unit uses the TAPI unit 240 to exchange data with the TMS unit 230. The TMS unit 230 is able to communicate with and manage one or more switches. For example, with reference to FIG. 1, the TMS unit 230 included in the server 110 can manage the switches 130A, 130B. Through the TAPI unit 240, the TMS unit 230 presents an application with a computer-telephony integration (CTI) view of these switches 130A, 130B. This allows the application to manage the switches 130A, 130B. Such switches 130A, 130B can operate without an associated TMS unit 230 if CTI features are not being used.

C. Application Unit Architecture

[0035] In the illustrated embodiment, the server 110 includes one application unit – account code (ACC) unit 250. In general, the ACC unit 250 handles a request to make a call, prompts for an account code, and completes the call. The functionality of the ACC unit 250 will be further described below with reference to FIG. 4.

[0036] In one embodiment, ACC unit 250 is implemented as a service that is used by the TMS unit 230. Communication or data exchange between the TMS unit 230 and the ACC unit 250 is further described with reference to FIG. 3. Although ACC unit 250 is illustrated as executing on the server 110, ACC unit 250 may be distributed among computing devices as is known to one of skill in the art. For example, the functionality enabled by ACC unit 250 may be implemented in a client-server fashion by having the client (user's local system) perform some functions and the server (ACC unit) perform others.

[0037] FIG. 3 is an illustration of one embodiment of the account code unit. Generally, ACC unit 250 includes several modules for receiving calls, prompting for account codes, and completing calls. In the illustrated embodiment, the ACC unit 250 includes an ACC control module 300, a call status module 310, an endpoint interface module 320, a storage interface module 330, a call transfer module 340, a TAPI interface module 350, a configuration module 360, and an extension library module 370. A data bus 390 communicatively couples each of the modules 300, 310, 320, 330, 340, 350, 360, 370.

[0038] The modules 300, 310, 320, 330, 340, 350, 360, 370 include program instructions that can be executed on, for example, processor 210 to implement the features or functions of the present invention. The modules 300, 310, 320, 330, 340, 350, 360, 370 are typically stored in a memory, such as memory 220. For server 110, the program instructions can be distributed on a computer readable medium or storage volume. The computer readable storage volume can be available via a public network, a private network, or the Internet. Program instructions can be in any appropriate form, such as source code, object code, or scripting code.

[0039] The ACC control module 300 centrally controls the operation and process flow of ACC unit 250, transmitting instructions and data to as well as receiving data from each module 310, 320, 330, 340, 350, 360, 370. Details of its operation will be discussed below with reference to FIG. 4.

[0040] The call status module 310 determines whether an account code is required or optional for a given call. In order to do this, the call status module 310 queries the TMS unit 230 through the TAPI interface module 350 and the TAPI unit 240. This gives the call status module 310 access to the call on the switch 130 itself. As previously described, the TAPI interface module 350 enables other modules to use the server's TAPI unit 240.

[0041] The endpoint interface module 320 sends data to and receives data from endpoints, such as analog phone 121A and IP phone 122, and devices running call manager software 150 that controls these endpoints. Endpoint interface module 320 communicates with a call manager 150 directly, while it communicates with an endpoint

directly or via a switch 130, depending on the type of the endpoint. For example, endpoint interface module 320 communicates with IP phone 122 directly but communicates with analog phone 121A through switch 130A.

[0042] Data sent to an endpoint or a call manager 150 comprises information such as, for example, whether an account code is required or optional and, if an account code has been received, whether the code is valid or invalid. Data received from an endpoint comprises information such as, for example, an account code to use for the phone call.

[0043] The storage interface module 330 accesses information stored in storage device 140. As described above, storage device 140 contains account code information, including account codes and the names of customers associated with the account codes.

[0044] The call transfer module 340 completes a phone call by transferring it to the call's destination phone number.

[0045] The configuration module 360 provides information about how the ACC control module 300 should operate. For example, the configuration module 360 determines how long the ACC control module 300 should wait for an account code to be entered before the ACC control module 300 times out and how many times an invalid account code may be entered before a phone call is terminated. Lastly, the extension library module 370 provides common functions that are used by the other modules.

D. Methods

[0046] Details of modules 300, 310, 320, 330, 340, 350, 360, 370 will be further explained with reference to FIG. 4. FIG. 4 illustrates a method performed by a server for

handling a request to make a call, prompting for an account code, and completing the call according to one embodiment of the present invention. The method 400 of FIG.4 would be performed once for each call coming into a site. In one embodiment, different calls are handled in parallel.

[0047] Before the method 400 of FIG. 4 begins, a switch 130A in the local site 100A receives a request to make a call. The request is associated with a user. The switch 130 determines whether account codes are enabled for the call. If account codes are not enabled, then the switch 130 processes the call normally. If account codes are enabled, then the switch 130 diverts the call to the server 110 and method 400 begins.

[0048] The ACC control module 300 determines 410 which devices (endpoints and call managers 150) are associated with the user who initiated the call. The ACC control module 300 then sends 420 a signal to each of these devices, using the endpoint interface module 320. These signals may be identical or they may differ based on the destination device. These signals cause the devices to prompt the user for an account code and, in one embodiment, indicate whether an account code is required or optional.

[0049] In traditional telephony systems, only the device that was used to initiate the call prompts the user for an account code. In contrast, in one embodiment, the ACC control module 300 sends signals to multiple devices, including devices that were not used to initiate the call, causing the devices to prompt the user for input. The ACC control module 300 then instructs the endpoint interface module 320 to wait 425 for either a reply from a device or a timeout, whichever occurs first. In one embodiment, a reply indicates which device generated it.

[0050] In this way, a user can initiate a call on one device and then use a different device to enter an account code. For example, a user may want to initiate a call on an analog phone 121 because he plans to use the phone during the conversation. Then, when the user realizes that he must enter an account code, he can use a call manager 150 to search for the appropriate account code using a software account code directory. Once he has located the account code, he can merely click on it using the call manager 150, rather than entering the code into the analog phone 121 digit by digit. As another example, a user may initiate a call using a call manager 150 and then enter an account code using an IP-based phone 122.

[0051] If the endpoint interface module 320 times out first 430, then the ACC control module 300 instructs the call status module 310 to determine 435 whether an account code is required for the call. If an account code is not required, then the ACC control module 300 processes 415 the call normally using the call transfer module 340 and the method ends. If an account code is required, then the method passes to step 455, which is described below.

[0052] If a reply is received first 440, then the ACC control module 300 determines 445 whether the received reply contains a valid account code. The ACC control module 300 does this by using storage interface module 330 to check whether the storage device 140 contains the received account code. If multiple account codes are received (e.g., one account count from each device that prompted the user), then the ACC control module 300 checks the validity of each one, in the order that is was received, until it has found a valid one. If the received account code is valid, then the ACC control module 300 attaches 450 the account code to the phone call and sends a signal to the devices (that are

associated with the user who initiated the call), using the endpoint interface module 320, that it no longer needs an account code. After a device has received this signal, it stops prompting the user for an account code. Then, the ACC control module 300 processes 415 the call normally using the call transfer module 340 and the method ends.

[0053] If the received account code is not valid, then the ACC control module 300 determines 455 whether the maximum number of attempts to enter a valid account code has been reached. If the maximum number has not been reached, then the method returns to step 420 and the ACC control module 300 again prompts 420 the devices for an account code using the endpoint interface module 320. If the maximum number has been reached, then the phone call is disconnected and the method ends.

[0054] Having described embodiments of a server with account code capability for distributed IP telephony systems (which are intended to be illustrative and not limiting), it is noted that modifications and variations can be made by persons skilled in the art in light of the above teachings. It is therefore to be understood that changes may be made in the particular embodiments of the invention disclosed that are within the scope and spirit of the invention as defined by the appended claims and equivalents.